WHAT IS CLAIMED IS:

1. A method comprising:

determining an authentication type to be used between a first node and a second node in a networked computer system;

plugging in a first authentication protocol handler module on the first node for the determined authentication type, wherein the first authentication protocol handler module is configured for use in generating authentication information for the first node for sending to the second node;

plugging in a second authentication protocol handler module on the second node for the determined authentication type, wherein the second authentication protocol handler module is configured for use in determining if the first node is authentic using the first node authentication information;

determining an access control model to be used by the second node in controlling access to resources of the second node by the first node; and

plugging in an access control context module for the determined access control model on the second node, wherein the access control context module is configured for use in controlling access to resources of the second node by the first node using the access control model.

2. The method as recited in claim 1, further comprising loading the determined access control model.

3. The method as recited in claim 1, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the second node by the first node.

4. The method as recited in claim 1, wherein the first authentication protocol handler module includes a handle request method, wherein the second authentication protocol handler module includes a handle response method, wherein the handle request method and handle response method are configured to exchange authentication information during an authentication process for the first node.

5. The method as recited in claim 1, further comprising:

the second node sending a challenge to the first node, wherein the challenge is in accordance with the determined authentication type;

the first authentication protocol handler module generating response data in response to the challenge, wherein the response data includes information for use in authenticating the first node;

the first node sending the response data to the second node; and

the second authentication protocol handler module authenticating the first node using the received response data.

6. The method as recited in claim 5, wherein said authenticating the first node using the received response data comprises:

the second authentication protocol handler module sending the received response data to a user repository, wherein the user repository comprises node information associated with one or more network nodes; and

the user repository comparing the response data to the node information to authenticate the first node.

7. The method as recited in claim 1, further comprising:

authenticating the first node using the first authentication protocol handler module and the second authentication protocol handler module;

the authenticated first node sending to the second node a request for access to a resource of the second node; and

the access control context module determining if the first node has access permission to the resource in response to the request for access to the resource of the second node.

8. The method as recited in claim 7, further comprising:

if said determining determines the first node has access permission to the resource, allowing the first node to access the resource; and

if said determining determines the first node does not have access permission to the resource, inhibiting the first node from accessing the resource.

9. The method as recited in claim 1,

wherein the second authentication protocol handler module is further configured for use in generating authentication information for the second node for sending to the first node; and

wherein the first authentication protocol handler module is further configured for use in determining if the second node is authentic using the second node authentication information.

10. The method as recited in claim 1, wherein the networked computer system is a messaging-based system.

11. The method as recited in claim 1, wherein the networked computer system uses the Java Message Service (JMS) to support messaging between nodes in the network.

12. The method as recited in claim 1, wherein networked computer system is a client-server system, wherein the first node is a client in the client-server system, and wherein the second node is a server in the client-server system.

13. The method as recited in claim 1, wherein the networked computer system is a peer-to-peer system, wherein the first node and the second node are peers in the peer-to-peer system.

14. A method for authenticating nodes in a networked computer system, comprising:

a first node initiating a connection to a second node in the networked computer system;

determining an authentication type to be used by the first node and the second node;

initializing a first authentication protocol handler on the first node for the determined authentication type;

initializing a second authentication protocol handler on the second node for the determined authentication type;

the second node sending a challenge to the first node, wherein the challenge is in accordance with the determined authentication type;

the first authentication protocol handler generating response data in response to the challenge, wherein the response data includes information for use in authenticating the first node;

the first node sending the response data to the second node; and

the second authentication protocol handler authenticating the first node using the received response data;

wherein the first authentication protocol handler and the second authentication protocol handler are pluggable modules configured to be replaced to support different authentication types.

15. The method as recited in claim 14, wherein said authenticating the first node using the received response data comprises:

the second authentication protocol handler sending the received response data to a user repository, wherein the user repository comprises node information associated with one or more nodes; and

the user repository comparing the response data to the node information to authenticate the first node.

16. The method as recited in claim 14, further comprising, if the first node is successfully authenticated:

determining an access control model to be used by the second node for the first node; and

initializing an access control context module for the determined access control model, wherein the access control context module is configured for use in controlling access to resources of the second node by the first node using the access control model.

17. The method as recited in claim 16, wherein the access control context module is a pluggable module configured to be replaced to support different access control models.

18. The method as recited in claim 16, wherein the access control context module is configured to support different pluggable access control models.

19. The method as recited in claim 16, further comprising loading the determined access control model.

20. The method as recited in claim 16, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the second node by the first node.

21. The method as recited in claim 16, further comprising:

the first node sending to the second node a request for access to a resource of the second node;

the access control context module determining if the first node has access permission to the resource;

if said determining determines the first node has access permission to the resource, allowing the first node to access the resource; and

if said determining determines the first node does not have access permission to the resource, inhibiting the first node from accessing the resource.

22. The method as recited in claim 14, wherein the networked computer system is a messaging-based system.

23. The method as recited in claim 14, wherein the networked computer system uses the Java Message Service (JMS) to support messaging between entities in the network.

24. The method as recited in claim 14, wherein networked computer system is a client-server system, wherein the first node is a client in the client-server system, and wherein the second node is a server in the client-server system.

25. The method as recited in claim 14, wherein the networked computer system is a peer-to-peer system, wherein the first node and the second node are peers in the peer-to-peer system.

26. A method comprising:

a second node determining an authentication type to be used by the second node to authenticate a first node in a networked computer system;

the second node plugging in a second authentication protocol handler module for the determined authentication type, wherein the second authentication protocol handler module is configured for use in determining if the first node is authentic using authentication information associated with the first node, wherein the first node authentication information is generated by a pluggable first authentication protocol handler module on the first node for the determined authentication type;

the second node determining an access control model to be used by the second node for the first node; and

the second node plugging in an access control context module for the determined access control model, wherein the access control context module is configured for use in controlling access to resources of the second node by the first node using the access control model.

27. The method as recited in claim 26, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the second node by the first node.

28. The method as recited in claim 26, wherein the second authentication protocol handler module includes a handle response method, wherein the handle response method is configured to exchange authentication information with a corresponding a handle request method of the first authentication protocol handler module during an authentication process for the first node.

29. The method as recited in claim 26, further comprising:

the second node sending a challenge to the first node, wherein the challenge is in accordance with the determined authentication type;

the second authentication protocol handler module receiving response data in response to the challenge, wherein the response data includes information for use in authenticating the first node, and wherein the response data is generated by the first authentication protocol handler module; and

the second authentication protocol handler module authenticating the first node using the received response data.

30. The method as recited in claim 29, wherein said authenticating the first node using the received response data comprises:

the second authentication protocol handler module sending the received response data to a user repository, wherein the user repository comprises node information associated with one or more nodes; and

the user repository comparing the response data to the node information to authenticate the first node.

31. The method as recited in claim 26, further comprising:

authenticating the first node;

the access control context module receiving a request for access to a resource of the second node from the authenticated first node; and

the access control context module determining if the first node has access permission to the resource in response to the request for access to the resource of the second node.

32. The method as recited in claim 31, further comprising:

if said determining determines the first node has access permission to the resource, the second node allowing the first node to access the resource; and

if said determining determines the first node does not have access permission to the resource, the second node inhibiting the first node from accessing the resource.

33. The method as recited in claim 26, wherein the networked computer system is a messaging-based system.

34. The method as recited in claim 26, wherein the networked computer system uses the Java Message Service (JMS) to support messaging between entities in the network.

35. The method as recited in claim 26, wherein networked computer system is a client-server system, wherein the first node is a client in the client-server system, and wherein the second node is a server in the client-server system.

36. The method as recited in claim 26, wherein the networked computer system is a peer-to-peer system, wherein the first node and the second node are peers in the peer-to-peer system.

37. A system comprising:

a first node comprising a first memory, wherein the first memory comprises first program instructions executable within the first node to initiate a connection request to the second node;

a second node comprising a second memory, wherein the second memory comprises second program instructions;

wherein the second program instructions are executable within the second node to:

determine an authentication type for use in authentication of the first node in response to the first program instructions initiating a connection request to the second node;

initialize a second authentication protocol handler module on the second node for the determined authentication type;

determine an access control model to be used by the second node; and

initialize an access control context module for the determined access control model, wherein the access control context module is configured for use in controlling access to resources of the second node by the first node using the access control model;

wherein the first program instructions are further executable within the first node to initialize a first authentication protocol handler module on the first node for the determined authentication type; and

wherein the first authentication protocol handler module and the second authentication protocol handler module are pluggable modules configured to be replaced to support different authentication types.

38. The system as recited in claim 37, wherein the access control context module is a pluggable module configured to be replaced to support different access control models.

39. The system as recited in claim 37, wherein the access control context module is configured to support different pluggable access control models.

40. The system as recited in claim 37, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the second node by the first node.

41. The system as recited in claim 37, wherein the second program instructions are further executable within the second node to:

send a challenge to the first node, wherein the challenge is in accordance with the determined authentication type;

wherein the first authentication protocol handler module is executable within the first node to generate response data in response to the challenge, wherein the response data includes information for use in authenticating the first node;

wherein the first program instructions are further configured to send the response data generated by the first authentication protocol handler module to the second node; and

wherein the second authentication protocol handler module is executable within the second node to authenticate the first node using the received response data.

42. The system as recited in claim 41, wherein the second node further comprises a user repository comprising information associated with one or more nodes, and wherein, in said authenticating the first node using the received response data, the second authentication protocol handler module is further executable within the second node to compare the response data received from the first node to the node information in the user repository to authenticate the first node.

43. The system as recited in claim 37,

wherein the second authentication protocol handler module is executable within the second node to exchange information with the first authentication protocol handler module executing within the first node to authenticate the first node;

wherein the first program instructions are further executable within the first node to send to the second node a request for access to a resource of the second node; and

wherein the access control context module is executable within the second node to determine if the first node has access permission to the resource in response to the request for access to the resource of the second node.

44. The system as recited in claim 43, wherein the second program instructions are further executable within the second node to:

allow the first node to access the resource if said determining determines the first node has access permission to the resource; and

inhibit the first node from accessing the resource if said determining determines the first node does not have access permission to the resource.

45. The system as recited in claim 37, wherein the system is a messaging-based system.

46. The system as recited in claim 37, wherein system uses the Java Message Service (JMS) to support messaging between the first node and the second node.

47. The system as recited in claim 37, wherein the system is a client-server system, wherein the second node is a server node, wherein the second program instructions are further executable within the second node to implement a server, and wherein the first node is a client node, wherein the first program instructions are further executable within the first node to implement a client application.

48. The system as recited in claim 37, wherein the system is a peer-to-peer system, wherein the first node and the second node are peers in the peer-to-peer system.

49. A system comprising:

a first node comprising a first memory, wherein the first memory comprises first program instructions executable within the client node to implement a client application;

a second node comprising a second memory, wherein the second memory comprises second program instructions executable within the second node to implement a server;

wherein the server is executable within the server node to:

receive a connection request from the client application;

determine an authentication type for use in authentication of the client application in response to the connection request;

plug in a server-side authentication protocol handler module for the determined authentication type;

wherein the client application is executable within the client node to plug in a client-side authentication protocol handler module for the determined authentication type;

wherein the client-side authentication protocol handler module is executable within the client node to:

receive a challenge from the server, wherein the challenge is in accordance with the determined authentication type;

generate response data in response to the received challenge, wherein the response data includes information for use in authenticating the client application;

wherein the server-side authentication protocol handler module is executable within the server node to:

receive the generated response data; and

authenticate the client application using the received response data.

50. The system as recited in claim 49, wherein the server node further comprises a user repository comprising client information associated with one or more clients, and

wherein, in said authenticating the client application using the received response data, the server-side authentication protocol handler module is further executable within the server node to compare the received response data to the client information in the user repository to authenticate the client.

51. The system as recited in claim 49, wherein, if the client is successfully authenticated, the server is further executable within the server node to:

determine an access control model to be used by the server for the client application; and

plug in an access control context module for the determined access control model, wherein the access control context module is configured for use by the server in controlling access to resources of the server by the client application.

52. The system as recited in claim 51, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the server by the client.

53. The system as recited in claim 51, wherein the access control context module is executable within the server node to:

receive a request for access to a resource of the server from the client application;

determine if the client application has access permission to the resource;

if said determining determines the client application has access permission to the resource, permitting the client application to access the resource; and

if said determining determines the client application does not have access permission to the resource, inhibiting the client application from accessing the resource.

54. The system as recited in claim 49, wherein the system is a messaging-based system.

55. The system as recited in claim 49, wherein the system uses the Java Message Service (JMS) to support messaging between entities in the system.

56. A server system comprising:

a memory, wherein the memory comprises program instructions executable within the server node to implement a server;

wherein the server is executable within the server node to:

receive a connection request from a client application;

determine an authentication type for use in authentication of the client application in response to the connection request;

plug in a server-side authentication protocol handler module for the determined authentication type; and

send a challenge to the client application, wherein the challenge is in accordance with the determined authentication type;

wherein the server-side authentication protocol handler module is executable within the server system to

receive response data from the client application, wherein the response data was generated by a pluggable client-side authentication protocol handler module in response to the challenge, wherein the response data includes information for use in authenticating the client application; and

authenticate the client application using the received response data.

57. The server system as recited in claim 56, wherein the server system further comprises a user repository comprising client information associated with one or more clients of the server, and wherein, in said authenticating the client application using the received response data, the server-side authentication protocol handler module is further executable within the server system to compare the received response data to the client information in the user repository to authenticate the client.

58. The server system as recited in claim 56, wherein, if the client is successfully authenticated, the server is further executable within the server system to:

determine an access control model to be used by the server for the client application; and

plug in an access control context module for the determined access control model, wherein the access control context module is configured for use by the server in controlling access to resources of the server by the client application.

59. The server system as recited in claim 58, wherein the access control context module encapsulates information configured for use in controlling access to the resources of the server by the client.

60. The server system as recited in claim 58, wherein the access control context module is executable within the server node to:

receive a request for access to a resource of the server from the client application;

determine if the client application has access permission to the resource;

if said determining determines the client application has access permission to the resource, permitting the client application to access the resource; and

if said determining determines the client application does not have access permission to the resource, inhibiting the client application from accessing the resource.

61. The server system as recited in claim 56, wherein the server system is a messaging-based system.

62. The server system as recited in claim 56, wherein the server system uses the Java Message Service (JMS) to support messaging between entities in the system.

63. A carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

a first node initiating a connection to a second node in a networked computer system;

determining an authentication type to be used by the first node and the second node;

initializing a first authentication protocol handler on the first node for the determined authentication type;

initializing a second authentication protocol handler on the second node for the determined authentication type;

the second node sending a challenge to the first node, wherein the challenge is in accordance with the determined authentication type;

the first authentication protocol handler generating response data in response to the challenge, wherein the response data includes information for use in authenticating the first node;

the first node sending the response data to the second node; and

the second authentication protocol handler authenticating the first node using the received response data;

wherein the first authentication protocol handler and the second authentication protocol handler are pluggable modules configured to be replaced to support different authentication types.

64. The carrier medium as recited in claim 63, wherein, in said authenticating the first node using the received response data, the program instructions are further computer-executable to implement comparing the response data to information comprised in a user repository, wherein the information is associated with one or more nodes.

65. The carrier medium as recited in claim 63, wherein, if the first node is successfully authenticated, the program instructions are further computer-executable to implement:

determining an access control model to be used by the second node for the first node; and

initializing an access control context module for the determined access control model, wherein the access control context module is configured for use in controlling access to resources of the second node by the first node using the access control model.

66. The carrier medium as recited in claim 65, wherein the access control context module is a pluggable module configured to be replaced to support different access control models.

67. The carrier medium as recited in claim 65, wherein the access control context module is configured to support different pluggable access control models.

68. The carrier medium as recited in claim 65, wherein the program instructions are further computer-executable to implement:

the first node sending to the second node a request for access to a resource of the second node;

the access control context module determining if the first node has access permission to the resource;

if said determining determines the first node has access permission to the resource, allowing the first node to access the resource; and

if said determining determines the first node does not have access permission to the resource, inhibiting the first node from accessing the resource.

69. The carrier medium as recited in claim 63, wherein networked computer system is a client-server system, wherein the first node is a client in the client-server system, and wherein the second node is a server in the client-server system.

70. The carrier medium as recited in claim 63, wherein the networked computer system is a peer-to-peer system, wherein the first node and the second node are peers in the peer-to-peer system.